

REMARKS

The Examiner is thanked for the telephone interview on July 21, 2003. On July 22, 2003, the Examiner agreed to withdraw the rejections of Claims 11-23, 34-46, and 57-105 under the first and second paragraphs of 35 U.S.C. §112.

Reconsideration of the application in view of the following remarks is respectfully requested. No claims have been added, cancelled, or amended. Claims 11-23, 34-46, and 57-105 are currently pending in the application.

103(a) REJECTION BASED ON GINTER, ELGAMAL, AND PRESSMAN

The Final Office Action rejected Claims 11-23, 34-46, and 57-105 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application Publication No. US 2002/0112171 A1 by Ginter, et al. ("Ginter") in view of U.S. Patent No. 6,389,534 B1 issued to Elgamal, et al. ("Elgamal") in view of Pressman, "Software Engineering: A Practitioner's Approach" (Pressman). The rejection is respectfully traversed.

CLAIM 11

With respect to independent method Claim 11, there is recited a method performed by a framework in a system comprising the framework, an application, and an implementation class that provides an implementation for a particular service; that method comprising:

receiving a request from an application for a customized implementation of a particular service;

instantiating an implementation class which provides an implementation for the particular service to give rise to an implementation instance;

determining a set of zero or more restrictions to be imposed on said customized implementation;

instantiating a wrapper class to give rise to a wrapper instance, said wrapper instance comprising enforcement logic for enforcing said restrictions;

encapsulating said implementation instance and said restrictions within said wrapper instance; and

providing said wrapper instance to the application as said customized implementation;

wherein **said wrapper instance comprises one or more invocable methods**, wherein **said implementation instance comprises one or more invocable methods**, and wherein encapsulating comprises:

**mapping the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance.**

(emphasis added).

The method of Claim 11 is quite advantageous because it provides an application with a wrapper instance as a customized implementation of a particular service. The wrapper instance encapsulates an implementation instance of the particular service, and also comprises enforcement logic that enforces a set of restrictions imposed on the implementation instance. **Because one or more invocable methods of the wrapper instance are mapped to one or more invocable methods of the implementation instance, an application to which the wrapper instance is provided may invoke the methods of the wrapper instance in the same manner as the application would have invoked the methods of the implementation instance.** Consequently, the application does not need to be modified to handle the wrapper instance differently than the

implementation instance. These and other benefits can be derived from the method of Claim 11.

### GINTER

Ginter neither teaches nor suggests such a method. Instead, Ginter discloses a content container object that contains information content and a permissions record. The information content may be, for example, text, sound, video, or computer software. The permissions record specifies rights associated with the content container object such as, for example, who can open the container object, who can use the container object's contents, and who can distribute the container object. An end user cannot use or access the container object's contents unless the permissions record has been delivered to the end user. In this way, the use of the information content may be restricted.

The method implemented in Ginter is quite different from the method claimed in Claim 11. One point to note is that, unlike Claim 11, Ginter does not map one or more invocable methods of a wrapper instance to one or more invocable methods of an implementation instance. Information content such as text, sound, and video does not comprise any invocable methods. Therefore, when the information content is text, sound, or video, there are no invocable methods of an implementation instance to which invocable methods of a wrapper instance could be mapped. Additionally, Applicants do not find any teaching or suggestion in Ginter that computer software information content comprises invocable methods.

Even if computer software information content was assumed, *arguendo*, to comprise invocable methods, Applicants still do not find any teaching or suggestion in Ginter that methods of the content container are mapped to methods of computer

software information content. Thus, unlike Claim 11, there appears to be no teaching or suggestion in Ginter of **mapping one or more invocable methods of a wrapper instance to one or more invocable methods of an implementation instance**. In fact, the Office Action does not allege that Ginter teaches or suggests this feature. Thus, Claim 11 is patentable over Ginter, taken individually.

### ELGAMAL

Elgamal also neither teaches nor suggests such a method. Instead, Elgamal discloses an application program calling a service module, and the service module, in turn, calling a policy filter that has been configured by a policy filter initialization module. Based on the determination of its policy filter, the service module returns to the application either an error (if the operation is not allowed) or a result of the operation requested by the application program (see col. 5, line 41-42, and also col. 6, lines 42-43).

An important point to note regarding Elgamal is that, unlike the method of Claim 11, Elgamal does not teach or suggest **mapping one or more invocable methods of a wrapper instance to one or more invocable methods of an implementation instance**. The service modules of Elgamal do not include their own instances of the cryptographic modules. Thus, the service modules of Elgamal cannot be considered to “wrap around” the cryptographic modules in the same manner that the wrapper instances of Claim 11 encapsulate implementation instances. Therefore, Elgamal fails to teach or suggest wrapper instances. Because Elgamal fails to teach or suggest a **wrapper instance**, it logically follows that Elgamal also fails to teach or suggest **mapping one or more invocable methods of a wrapper instance** to one or more invocable methods of an

implementation instances. In fact, the Office Action does not allege that Elgamal teaches or suggests this feature. Thus, Claim 11 is patentable over Elgamal, taken individually.

#### PRESSMAN

Pressman also neither teaches nor suggests such a method. Instead, Pressman discloses general principles of object-oriented software engineering.

It does not appear that Pressman teaches or suggests **mapping one or more invocable methods of a wrapper instance to one or more invocable methods of an implementation instance** as recited by Claim 11. In fact, the Office Action does not allege that Pressman teaches or suggests this feature. Thus, Claim 11 is patentable over Pressman, taken individually.

Even assuming, arguendo, that it would have been obvious to combine Ginter, Elgamal, and Pressman, the combination of Ginter, Elgamal, and Pressman still fails to teach or suggest “**mapping the one or more invocable methods of said wrapper instance to the one or more invocable methods of said implementation instance**” as recited by Claim 11. Accordingly, Applicants submit that Claim 11 is patentable over Ginter and Elgamal.

#### CLAIMS 34 AND 57

Claims 34 and 57 are directed to a framework that comprises mechanisms for performing the method of Claim 11, and a computer-readable medium that contains instructions for implementing a framework for performing the method of Claim 11, respectively. Thus, Applicants submit that Claims 34 and 57 are patentable over Ginter,

Elgamal, and Pressman for at least the same reasons as those give above in connection with Claim 11.

#### REMAINING DEPENDENT CLAIMS

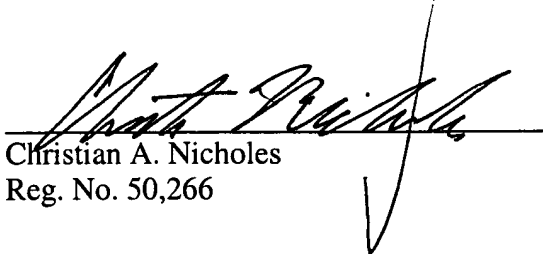
The pending claims not discussed so far are dependent claims that depend on an independent claim that is discussed above. Because each of the dependent claims includes the limitations of claims upon which it depends, the dependent claims are patentable for at least those reasons the claims upon which the dependent claims depend are patentable. Withdrawal of the rejections with respect to the dependent claims and allowance of the dependent claims is respectfully requested. In addition, the dependent claims introduce additional limitations that independently render them patentable. Due to the fundamental difference already identified, a separate discussion of those limitations is not included at this time.

For at least the reasons set forth above, Applicants respectfully submit that all pending claims are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all claims is hereby respectfully solicited.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: July 23, 2003

  
Christian A. Nicholes  
Reg. No. 50,266

1600 Willow Street  
San Jose, California 95125-5106  
Telephone No.: (408) 414-1080  
Facsimile No.: (408) 414-1076

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22513-1450

on July 23, 2003 by Trudy Bagdon  
(Date) (Signature)